

A method of 3D GUI design for SCADA using Unity

Kum-Thap Jang¹, Kyong-Mi Kim¹, Song-Hyon Kim¹, Kum-Song Ri*

¹Faculty of Mechanical Science and Technology, Kim Cheak University of Technology, Kyogudong No.60, Yonggwang street, Pyongyang 950003, Democratic People's Republic of Korea



Published in
VOI- 1 Issue: 5

DOI:10.5281/zenodo.17681295

PP: 23-38

*Correspondence:

Kum-Song Ri

Faculty of Mechanical Science
and Technology, Kim Cheak
University of Technology,
Kyogudong No.60, Yonggwang
street, Pyongyang 950003,
Democratic People's Republic
of Korea

Email: rks90531@star-co.net.kp

Abstract

With the recent advancement of science and technology, the large-scale SCADA system 2D GUI of large-scale production processes has been designed with more intuitive and efficient 3D GUI. This paper presents a method to design 3D GUI for SCADA of a process control system composed of PLC with Unity software for various applications including game development. That is, in a production process automation system, the information from the PLC deployed on site is database-based on the control room computer and this database is connected to the 3D GUI created by Unity to construct the SCADA system. The 3D GUI design for SCADA by Unity, which also meets ergonomic requirements, is very effective in the field of systems where the amount of data to be displayed on the interface is large and the complexity and detail must be represented. Then, we demonstrate the advantages of the 3D interface introduced by this method through user experiments.

Keywords: SCADA, unity, 3D GUI, PLC

Introduction

As everything is integrated and AI-enabled by information devices, including computers, real-time data processing is increasing.

As the production processes are integrated and expanded, a high-performance computer for big data processing is required.

However, with the rapid development of computer manufacturing, high-performance computers, which were not used at high prices in the past, have become popular among people because of their low cost.

This has provided the possibility to switch from SCADA systems that rely on 2D GUIs to 3D GUIs with higher visual and effective performance in the field of interface design in the process area where big data are processed.

The SCADA support program, including WinCC, Adroit and Lview, has a drawback in monitoring and data processing because 2D GUI is effective in low data throughput systems, but limited in large systems to reflect large amounts of data on interfaces [1]. To overcome this, the use of 3D graphics to visualize large amounts of data has been investigated [1-4].



GUI design Unity has powerful Graphics and UI functions, so GUI functionality implementation is not a problem. We have built a 3D model of the process plant with a 3D max or Bs SolidWorks application and imported it into Unity. The extension is *.fbx.

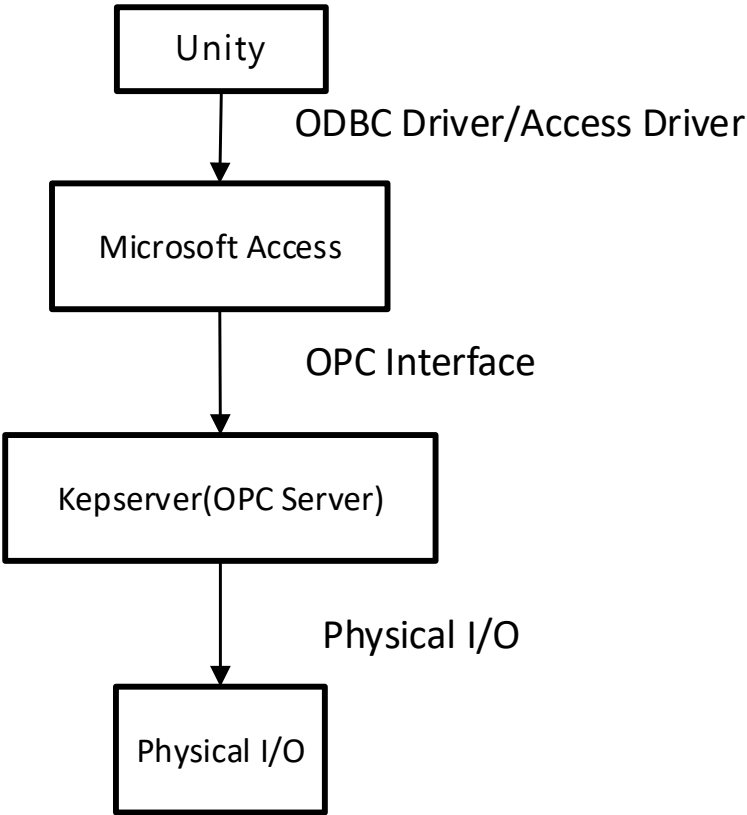


Fig 2. SCADA Design Architecture with Unity

We transfer to the Database System (database system) through Physical Interface (OPC interface) at physical I/O(Fig. 2).

The database system that receives data transfers data to Unity using database-specific tools, including ODBC Driver (Open Database Connectivity Driver) or Access Driver.

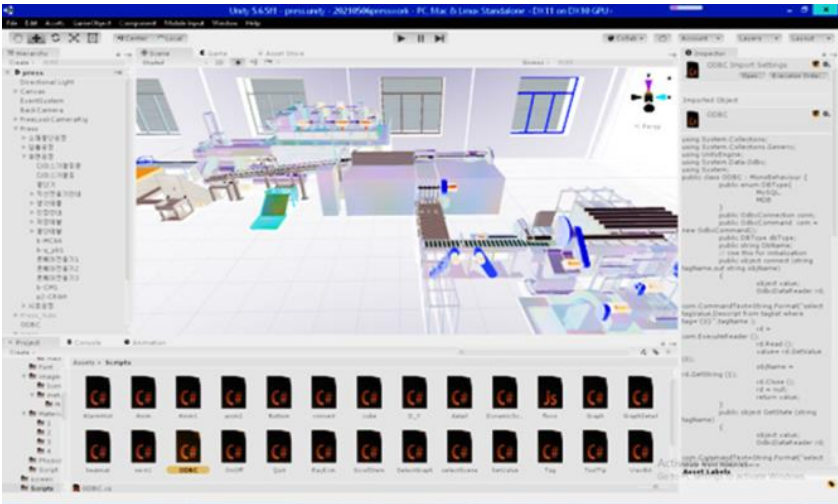


Fig 3. Unity Interface for Database Connection

The data link between Unity and Microsoft Access database is implemented using Unity's C# Script.

In Unity, create an object for database connectivity and connect it to a program named ODBC.cs (C# Script)

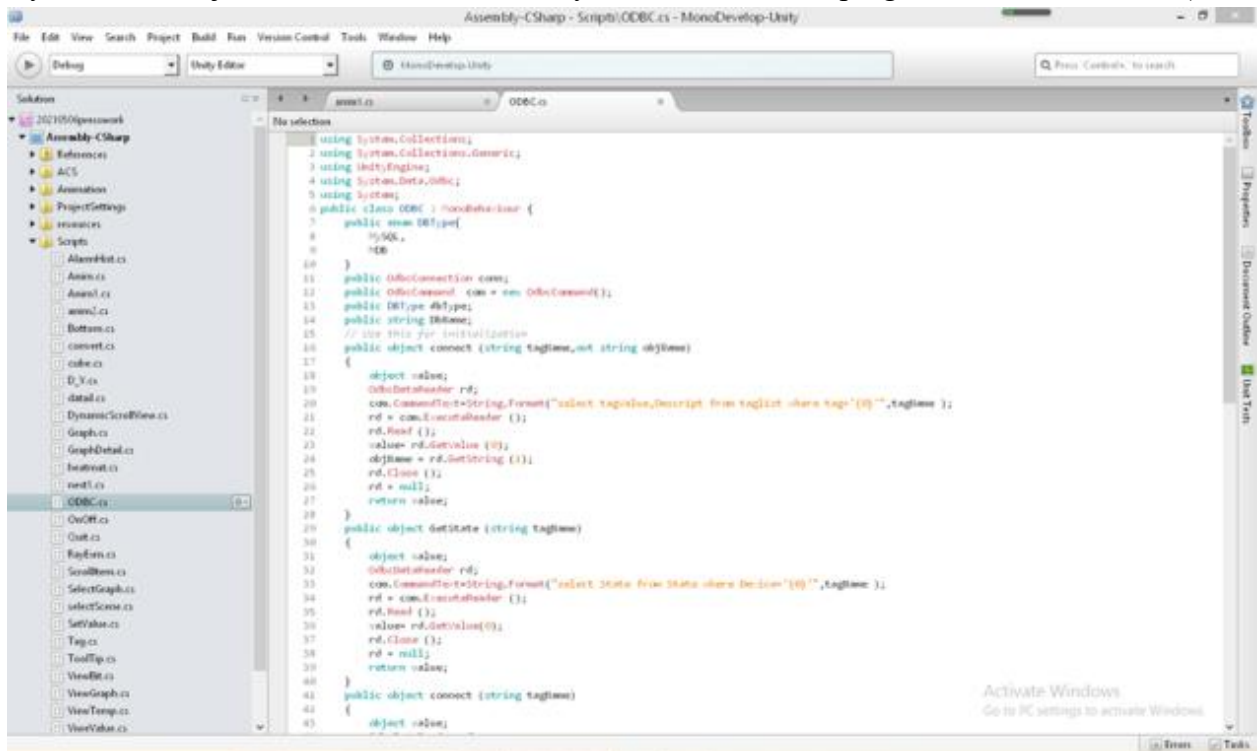


Fig 4. C# Script for connecting Unity and ODBC

Unity and Database Linking program flowchart(Fig.5).

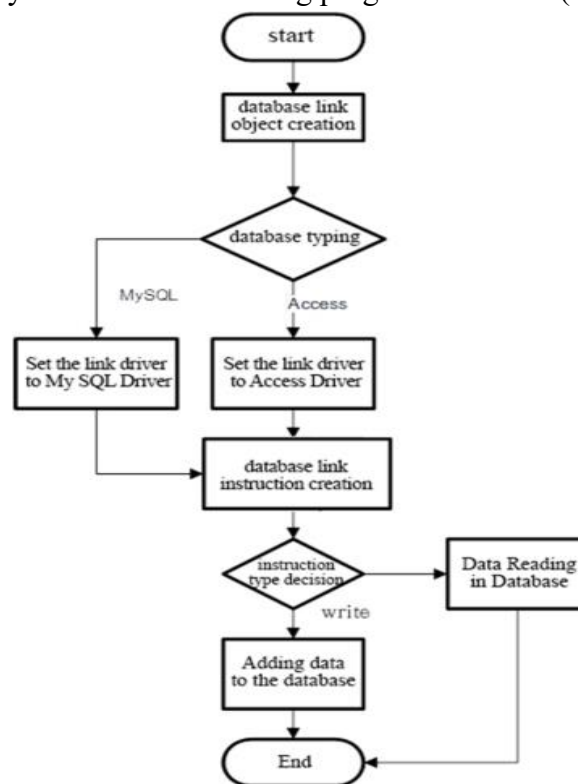


Fig 5. Unity and Database Linking Program Flow

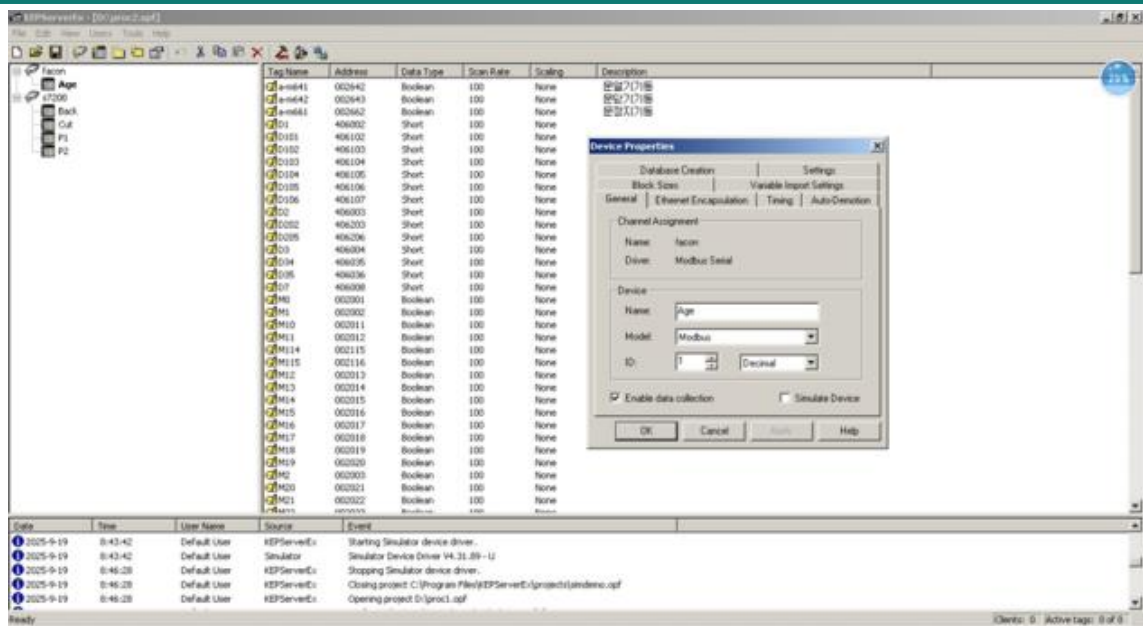


Fig 8. Address of process using KEPServer Ex V4.0.

Results and discussion

The evaluation of 3D GUI for SCADA of a Unity-based heat treatment process control system is done by indicating the advantages and disadvantages of users using the previous 2D GUI.

We set up the same task with the same data for 2D GUI and 3D GUI and measure and evaluate the time required to complete the task.

The object is inspected through the monitoring of the condition of the hot wire during the heat treatment process. The hot wire is in the heat treatment furnace and consists of 24 beams.

The process values to be monitored are temperature, voltage, current, frequency, etc.

Since the number of coil bundles of each heating wire is different, the monitoring data values are also individual.

We prepare 10 trained people to compare and evaluate 3D GUI and 2D GUI.

Five of the ten patients were male and five were female.

We choose the task but exclude the special case or the case of an abnormal problem.

The task is to find the maximum and minimum temperature values for 24 heating wires. The temperature range of the heat-treatment furnace is 0 to 580 ° C, and when the temperature is above 300 ° C, a yellow flame is created, when the temperature is above 500 ° C, a black flame is created. The performance of the proposed project is measured and tabulated by measuring the response time of 10 persons.

To ensure the accuracy of the task, the amount of data is tested according to 20, 50, 100, 200 and 300.

The color change of the filament with temperature change in 3D GUI(Fig. 8).

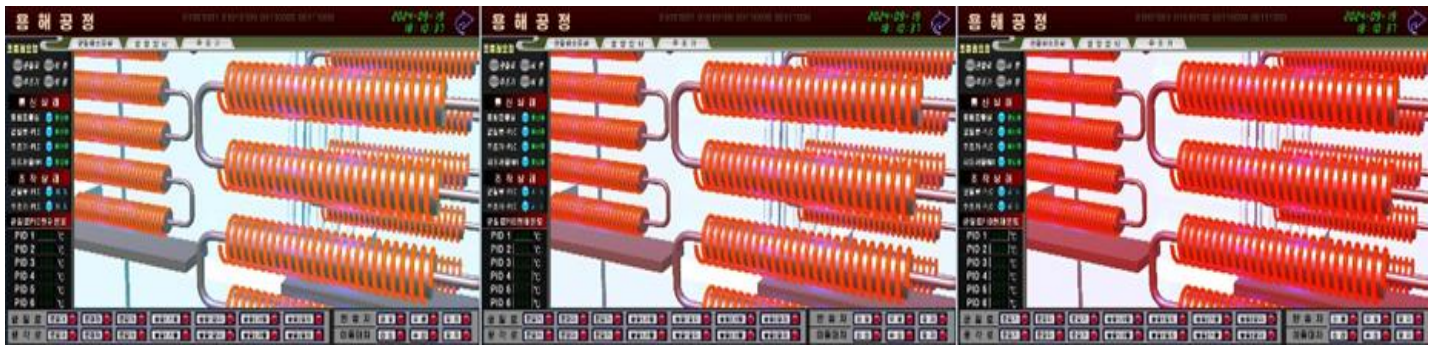


Fig.9. Heating wire visually showing the temperature change in Unity

The evaluation results for the task are as follows(Fig.9).

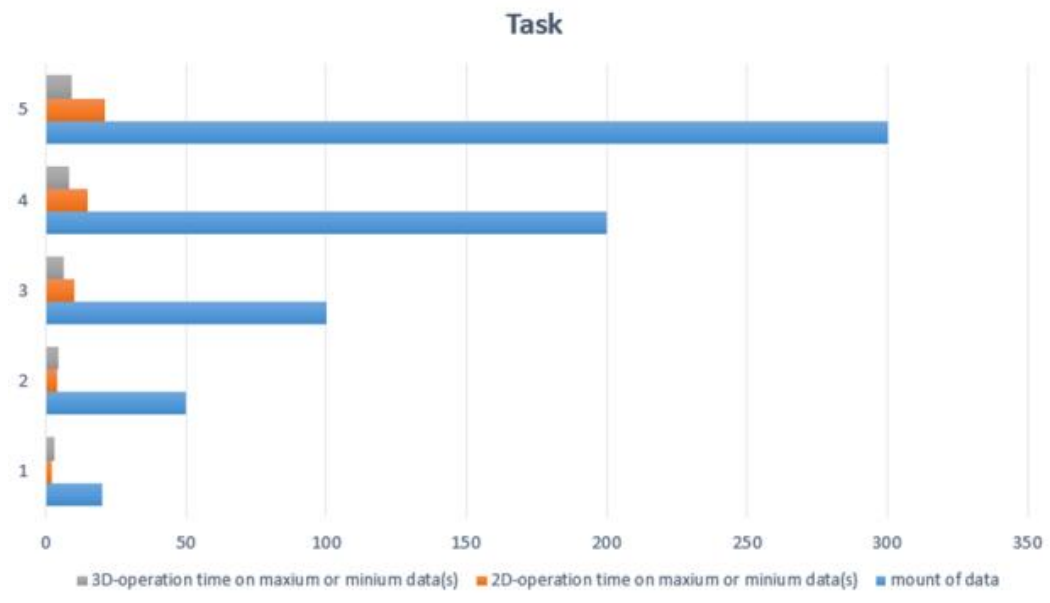


Fig.10. Evaluation results for the task

Conclusions

Applications that can be imported into Unity's graphics resources, including 3Ds max, are used in SCADA design by creating graphics that are similar to the real ones of complex mechanical equipment.

Simple and approximate designed patterns with the most important actuators can be connected by means of an electronic device installed in the actuators of the real plant and a sensor to address value correspondence by PLC software.

The 3D GUI of SCADA system, written in the Uniy program by OPC, communicates with PLC in real time.

This allows the real-time monitoring and control of large systems that require processing a large amount of information, both visually and stereo-conveniently.

In addition, the addition of information with the expansion of production will not affect the GUI configuration and design of the SCADA system.

This is another advantage of 3D GUI design for SCADA by Unity.

Furthermore, user experiments have shown that overall trends can be effectively recognized in 3D GUI when the amount of information is relatively large or not all information that needs detailed data can be displayed on a single screen with a moderate scale in 2D GUI.

However, the complexity of equipment graphics and real-time processing is highly dependent on computer performance, which is more expensive than 2D GUI, and the effort and time consuming of SCADA design.

The validity of these ideas should also be evaluated.

Data availability

All data used in this study are available with the approval of the authors.

The impact of the research topic

This research topic does not conflict with other published researchers.

Acknowledgement

I would like to thank the State Commission of Science and Technology of the DPRK for giving selfless assistance to our research work.

Reference

VERSION 5.00

Begin VB.Form Form1

BorderStyle = 1 'Fixed Single

Caption = "Form1"

ClientHeight = 1980

ClientLeft = 150

ClientTop = 795

ClientWidth = 3705

Icon = "Form.frx":0000

LinkTopic = "Form1"

MaxButton = 0 'False

MinButton = 0 'False

ScaleHeight = 1980

ScaleWidth = 3705

StartPosition = 3 'Windows Default

Begin VB.CommandButton Command2

Caption = "Contact off"

Height = 375

Left = 2040

TabIndex = 6

Top = 1440

Width = 1455

End

Begin VB.PictureBox pichook

Height = 975

Left = 5760

```
ScaleHeight = 915
ScaleWidth  = 1515
TabIndex    = 5
Top         = 0
Width       = 1575
End
Begin VB.CommandButton Command1
Caption     = "Contact"
Height     = 375
Left       = 240
TabIndex   = 4
Top        = 1440
Width      = 1455
End
Begin VB.TextBox ProcID
Height     = 285
Left       = 1440
TabIndex   = 3
Text       = "2"
Top        = 840
Width      = 1455
End
Begin VB.TextBox Server
Height     = 285
Left       = 1440
TabIndex   = 0
Text       = "192.168.1.3"
Top        = 360
Width      = 1455
End
Begin VB.Timer Timer1
Enabled    = 0 'False
Interval   = 1000
Left       = 360
Top        = 1440
End
Begin VB.Image Img
Height     = 240
Index      = 0
Left       = 4320
Picture    = "Form.frx":000C
Top        = 1560
Width      = 240
```

```
End
Begin VB.Image Img
    Height      = 240
    Index       = 1
    Left        = 5040
    Picture     = "Form.frx":0396
    Top         = 1560
    Width       = 240
End
Begin VB.Image Img
    Height      = 240
    Index       = 2
    Left        = 5760
    Picture     = "Form.frx":0720
    Top         = 1560
    Width       = 240
End
Begin VB.Label label2
    Caption     = "Process number"
    Height      = 255
    Left        = 240
    TabIndex    = 2
    Top         = 840
    Width       = 855
End
Begin VB.Label Label1
    Caption     = "Server address"
    Height      = 255
    Left        = 240
    TabIndex    = 1
    Top         = 360
    Width       = 1095
End
Begin VB.Menu mnupopup
    Caption     = "Menu"
    Begin VB.Menu mnuset
        Caption  = "Setting"
    End
    Begin VB.Menu mnuexit
        Caption  = "Finish"
    End
End
End
End
```

```
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Public Chanel As String
Public Device As String
Dim WithEvents ConnectedGroup As OPCGroup
Attribute ConnectedGroup.VB_VarHelpID = -1
Dim ConnectedOPCServer As OPCServer
Dim ConnectedServerGroup As OPCGroups
Dim OPCItemCollection As OPCItems
Dim ItemServerHandles() As Long
Dim ItemServerErrors() As Long
Dim ItemCount As Long
Dim conn As Connection
Dim conn1 As Connection
Dim rs As Recordset, rs1 As Recordset
Dim rec As Boolean
Dim PC As Boolean
Dim ID As String
Private Declare Function RegCreateKey Lib "advapi32.dll" Alias "RegCreateKeyA" (ByVal hKey As Long,
ByVal lpSubKey As String, phkResult As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32.dll" (ByVal hKey As Long) As Long
Private Declare Function RegSetValueEx Lib "advapi32.dll" Alias "RegSetValueExA" (ByVal hKey As
Long, ByVal lpValueName As String, ByVal Reserved As Long, ByVal dwType As Long, lpData As Any,
ByVal cbData As Long) As Long
Const HKEY_CURRENT_USER = &H80000001
Const HKEY_LOCAL_MACHINE = &H80000002
Private Type NOTIFYICONDATA
    cbSize As Long
    hWnd As Long
    uId As Long
    uFlags As Long
    ucallbackMessage As Long
    hIcon As Long
    szTip As String * 64
End Type
Private Const NIM_ADD = &H0
Private Const NIM_MODIFY = &H1
Private Const NIM_DELETE = &H2
Private Const NIF_MESSAGE = &H1
```

```
Private Const NIF_ICON = &H2
Private Const NIF_TIP = &H4
Private Const WM_LBUTTONDOWNBLCLK = &H203
Private Const WM_LBUTTONDOWN = &H201
Private Const WM_RBUTTONUP = &H205
Private Declare Function Shell_NotifyIcon Lib "shell32" Alias "Shell_NotifyIconA" (ByVal dwMessage As Long, pnid As NOTIFYICONDATA) As Boolean
Dim TrayI As NOTIFYICONDATA
Private Sub OPCCConnect()
    Dim OPCItemIDs() As String
    Dim ClientHandles() As Long
    SaveString HKEY_CURRENT_USER, "software\OPCclient", "a", Str(PC)
    conn.Execute "delete * from setValue"
    conn.Execute "Update State set State=0 where Device='c'"
    rs.Open "SELECT * FROM TagList", conn
    rs.MoveFirst
    Do Until rs.EOF
        ItemCount = rs!ID
        rs.MoveNext
    Loop
    ReDim OPCItemIDs(ItemCount) As String
    ReDim ClientHandles(ItemCount) As Long
    rs.MoveFirst
    Do Until rs.EOF
        OPCItemIDs(rs!ID) = rs!channel & "." & rs!Device & "." & rs!address & "@" & rs!DataType
        ClientHandles(rs!ID) = rs!ID
        rs.MoveNext
    Loop
    rs.Close
    Set ConnectedOPCServer = New OPCServer
    ConnectedOPCServer.Connect "KEPware.KEPServerEx.V4"
    Set ConnectedServerGroup = ConnectedOPCServer.OPCGroups
    ConnectedServerGroup.DefaultGroupIsActive = True
    ConnectedServerGroup.DefaultGroupDeadband = 0
    Set ConnectedGroup = ConnectedServerGroup.Add("group1")
    ConnectedGroup.UpdateRate = 1000
    ConnectedGroup.IsSubscribed = True
    Set OPCItemCollection = ConnectedGroup.OPCItems
    OPCItemCollection.DefaultIsActive = True
    OPCItemCollection.AddItems ItemCount, OPCItemIDs, ClientHandles, ItemServerHandles,
    ItemServerErrors
    TrayI.cbSize = Len(TrayI)
    TrayI.hWnd = pichook.hWnd
```

```
TrayI.uId = 1&
TrayI.uFlags = NIF_ICON Or NIF_TIP Or NIF_MESSAGE
TrayI.ucallbackMessage = WM_LBUTTONDOWN
TrayI.hIcon = Img(0).Picture
Shell_NotifyIcon NIM_ADD, TrayI
Me.Hide
End Sub
Public Sub OPCDisconnect()
Timer1.Enabled = False
conn.Close
Set rs = Nothing
Set conn = Nothing
OPCItemCollection.Remove ItemCount, ItemServerHandles, ItemServerErrors
Set OPCItemCollection = Nothing
ConnectedServerGroup.Remove ("group1")
Set ConnectedServerGroup = Nothing
Set ConnectedGroup = Nothing
ConnectedOPCServer.Disconnect
Set ConnectedOPCServer = Nothing
End Sub
Public Sub OPCItemWrite(ByVal Index As Integer, ByVal Value As Variant)
Dim item As OPCItem
Set item = OPCItemCollection.GetOPCItem(Val(ItemServerHandles(Index)))
item.Write (Val(Value))
End Sub
Public Function OPCItemQuality(ByVal Index As Integer)
Dim item As OPCItem
Set item = OPCItemCollection.GetOPCItem(Val(ItemServerHandles(Index)))
OPCItemQuality = item.Quality
End Function
Private Sub Command1_Click()
On Error GoTo Err_1
Timer1.Enabled = True
rec = False
Set conn = New Connection
Set conn1 = New Connection
Set rs = New Recordset
Set rs1 = New Recordset
'Driver=MySQL ODBC 5.1 Driver;SERVER=127.0.0.1;UID=root;DATABASE=proc2;PORT=3306
conn.Open "Driver=MySQL ODBC 5.1 Driver;SERVER=" & Server.Text &
";UID=root;DATABASE=proc" & ProcID.Text & ";PORT=3306"
OPCConnect
PC = True
```

```
Exit Sub
Err_1:
conn.Open "Driver={Microsoft Access Driver (*.mdb)};dbq=e:/database.mdb"
OPCConnect
PC = False
MsgBox " Access ", vbInformation, "Inform"
End Sub
Private Sub Command2_Click()
OPCDisconnect
End Sub
Sub ConnectedGroup_DataChange(ByVal TransactionID As Long, ByVal NumItems As Long,
ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As Date)
On Error GoTo Err
Dim i As Integer
For i = 1 To NumItems
Dim state As Boolean
state = False
If Qualities(i) = &HC0 Then state = True
conn.Execute "update TagList set tagValue=" & CCur(ItemValues(i)) & ",Enable=" & state & " where
id=" & ClientHandles(i)
If rs.state = ObjectStateEnum.adStateOpen Then rs.Close
rs.Open "select alarm,device from taglist where id=" & ClientHandles(i), conn
If CBool(rs!alarm) And CBool(ItemValues(i) And Val(rs!alarm) <> 3) Then
conn.Execute "insert into alarm (id,device,alarm,saved) values (" & ClientHandles(i) & "," &
rs!Device & "," & rs!alarm & ",now())"
End If
rs.Close
Next
Exit Sub
Err:
OPCDisconnect
Command1_Click
End Sub
Private Sub mnuset_Click()
Me.Show
End Sub
Private Sub Timer1_Timer()
Dim rs As Recordset, rs1 As Recordset
Set rs = New Recordset
Set rs1 = New Recordset
rs.Open "select * from setValue", con
If rs.EOF = False Then
rs.MoveFirst
```

```
rs1.Open "select id from Taglist where tag='" & rs!Tag & "'", conn
OPCItemWrite Int(rs1!ID), rs!m_value
rs1.Close
conn.Execute "delete * from setValue where ID='" & rs!ID
End If
rs.Close
conn.Execute "Update State set State='" & PC & " where Device='c'"
conn.Execute "Update State set State='" & OPCItemQuality(1) & " where Device='plc3'"
conn.Execute "Update State set State='" & OPCItemQuality(138) & " where Device='plc4'"
conn.Execute "Update State set State='" & OPCItemQuality(189) & " where Device='plc5'"
Dim n As Integer
Dim fld As Field
Dim tags As String
Dim values As String
n = n + 1
If n = 60 Then
rs.Open "SELECT * FROM hist", conn
rs.MoveFirst
For Each fld In rs.Fields
If fld.Name <> "Saved" Then
rs1.Open "SELECT id FROM TagList where tag='" & fld.Name & "'", conn
Print rs1!ID
If OPCItemCollection.item(Int(rs1!ID)).Quality Then
tags = tags & fld.Name & ","
values = values & OPCItemCollection.item(Int(rs1!ID)).Value & ","
End If
rs1.Close
End If
Next
conn.Execute "insert into hist (" & tags & "Saved) values (" & values & "Now())"
n = 0
End If

If PC Then
Static Tek As Integer
TrayI.hIcon = Img(Tek).Picture
Tek = Tek + 1
If Tek = 3 Then Tek = 1
Shell_NotifyIcon NIM_MODIFY, TrayI
TrayI.szTip = "Contact on" & Chr(0)
Else
TrayI.hIcon = Img(0).Picture
Shell_NotifyIcon NIM_MODIFY, TrayI
```

```
TrayI.szTip = "Contact off" & Chr(0)
End If
Set rs = Nothing
Set rs1 = Nothing

End Sub

Private Sub Form_Unload(Cancel As Integer)
    OPCDisconnect
    TrayI.cbSize = Len(TrayI)
    TrayI.hWnd = pichook.hWnd
    TrayI.uId = 1&
    Shell_NotifyIcon NIM_DELETE, TrayI
End
End Sub

Sub SaveString(hKey As Long, strPath As String, strValue As String, strData As String)
    Dim Ret
    RegCreateKey hKey, strPath, Ret
    RegSetValueEx Ret, strValue, 0, 1, ByVal strData, Len(strData)
    RegCloseKey Ret
End Sub

Private Sub mnuxit_Click()
    Unload Me
End Sub

Private Sub pichook_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim msg
    msg = X / Screen.TwipsPerPixelX
    If msg = WM_LBUTTONDOWN Then mnupopup.Visible = False
    If msg = WM_RBUTTONDOWN Then Me.PopupMenu mnupopup
End Sub
```